

FOR TIMEX/SINCLAIR ENTHUSIASTS

PUBLISHED BY WMJ DATA SYSTEMS

* * Volume 2, Number 3 * *

From the Desk of Bill Johnson

The WINTER 1987 issue which will be published in the early part of January 1987 will be our last issue. Due to a large increase in my business, WMJ Data Systems, I will not have the time needed to keep publishing QuarTerS.

I had many plans for QuarTerS, as discussed in the Summer 1986 issue, including an extra issue and a Best of QuarTerS. The extra business has come from my jump into the Apple, Atari, Commodore, IBM...lines of computers. Currently I need all of my time just to handle the every day events of running WMJ Data Systems. My free time is precious to me as I am sure yours is to you. We will still be handling products for the Timex/Sinclair line and plan to for as far as we can see. Many of you should have received our latest catalog recently, if not let us know.

If the upper right hand corner of your mailing label says FALL-86 than this is your last issue. If you wish to receive the WINTER 1987 issue please send a check or money order for \$2.50 by December 31st along with your name and address.

Since most of our subscriptions expire with the FALL 1986 or WINTER 1987 issues we will have fulfilled our obligation to most of our readers. For all other readers who have subscriptions that expire after the WINTER 1987 issue we will settle with you at the time the WINTER 1987 issue is published.

Currently the largest and most consistant Timex/Sinclair magazine is TIME DESIGNS, 29722 Hult Rd, Colton, OR 97017. Time Designs is published bi-monthly and a one year subscription goes for \$15.00. I recommend this magazine as it covers all aspect of the Timex/Sinclair computers.

Until next time...

OWN ANOTHER COMPUTER???

Do you have an...

APPLE

ATARI

COMMODORE

IBM PC/PCjr

IBM COMPATIBLE

TIMEX/SINCLAIR

TRS-80

IF YOU OWN ONE OF THE ABOVE COMPUTERS
WHY NOT SEND FOR A COPY OF OUR LATEST
CATALOG, 28 PAGES OF DISCOUNT SOFTWARE.

WMJ DATA SYSTEMS
4 Butterfly Drive
Hauppauge, NY 11788
(516)543-5252

DEAR QTS:

You might like to know that I run my office on a ZX81 computer, and will reprint a short article that was published in "M. D. Computing" here:

"DAYSHEET"

As 512K RAM/10 megabyte disk computer systems become more common-place, we sometimes forget what can be done with older, more common, 64K, 280 based 8 bit machines, and efficient programming.

Larry Sheingorn, M. D., of Rockville, Maryland, uses a \$44.00 Sinclair ZX81 computer to run his clinical ophthalmology practice. With additional memory, keyboard, and 80 column printer, the system cost approximately \$1,000.00. Lack of disks prevents large amounts of data being "on-line", but the program will soon be modified to take advantage of other, disk based systems, including CP/M, Commodore, Apple, and IBM devices. The cost of fully configured systems will be from 1/5th to 1/10th of those currently on the market.

The program essentially creates a file called "daysheet"; disk-based versions will store multiple files in RAM &/or disk. The "day" consists of up to 25 patient encounters, and each encounter can have up to five procedures. A "day" can also have up to 25 "ROA"'s (received on accounts), and daily expenses are grouped in 20 different categories.

A menu oriented structure allows an untrained operator to use the computer almost immediately. Obviously, the user is completely unaware of updating "files", "records", or "fields". A day's appointments are generally entered into the device at a convenient time, and the day is "SAVE"'d on magnetic tape. As the day progresses, appointments and other transactions are made in real time. At any time, a printed schedule, properly ordered, as well as chart forms for all "new" patients can be prepared with appropriate, single, keystrokes. Insurance ready bills are generated

with a keystroke for one or all patients, using data entered during the day, without having to consult CPT or ICD code books. Patient educational materials are also available thru simple responses to various menus.

Keep in touch. Its too bad the ZX81 died!

Sincerely,

Larry Sheingorn, M.D.
Rockville, MD 20850

Dear QTS:

I read with interest the Summer issue of QuarTerS and the letter from Joan Kealy, asking about the TS2068 in business prompted me to write.

I bought my TS2068 computer, 2020 cassette recorder and 2040 32 column printer on sale more than 2 years ago, just before Timex Corp. gave up on computers. I had planned to use the computer system in business, as I was just starting out as a self-employed bookkeeper and freelance writer. During the past two years, I have purchased some business software and replaced the TS2040 with a full-size 80 column Panasonic KXP-1091 Dot Matrix printer with NLQ mode, using an Aerco interface. I've had terrific results using the computer system in my freelance writing - the NLQ manuscripts look very professional and there are some very good word processing programs available for the TS2068. I presently use the Textwriter 2000 Plus Word Processing program but I've recently purchased the 64 Column MSCRIPT and hope to master the use of it soon. I really haven't had any trouble saving manuscripts on cassette tapes although it is time-consuming compared to disks.

I wish I could be as positive about the accounting possibilities on the TS2068. It is nearly impossible to find any integrated accounting software for the TS2068, and the main problem I've encountered is that most of the business accounting software (Payroll, AR, AP, Inventory) that is

available from 3rd party vendors do not support full-sized printers. They are only compatible with the old TS2040 thermal printer. The same is true of the original software produced by Timex - I have several programs Budgeter, VuCalc, Home Finance, etc., that I can't get printouts from because I've retired the TS2040. This is very frustrating and could be another reason the Timex failed in selling enough computers. It's too bad that when the TS2068's were originally developed, they weren't developed with the disk drive, 80 column printer and appropriate business software compatibility. It's almost as if the computer wasn't really regarded as a serious machine. I like using the 2068 but to add a disk drive at this stage would involve a lot of money and would there be any software? With most name brand computer prices falling fast, buying a new computer system with disk drives and available software would probably be a better choice. I'd hate to give up my TS2068 but there doesn't seem to be many alternatives. Anyway, I haven't been able to use the TS2068 very successfully for accounting applications and so I use it almost exclusively for word processing. It's been a relatively inexpensive way to become familiar with word processing.

Basically, I guess I'm biding time until and if 3rd party vendors develop more accounting software or until I can afford another computer system. I guess that is blasphemy to all dedicated TS users, but I'm getting tired of hassles. I'm looking forward to reading other letters from TS users who do use the TS2068 in their businesses. If anyone knows of any compatible business software, I'd welcome the information.

Sincerely,

Carolyn Bower
560 Catalina Ave
Wooster, OH 44691

Dear QTS:

Enclosed find a program that your readers with disk systems may find useful. This program automatically loads your program from the directory listed on the screen. I am using the AERCO disk system, but the variables may be changed to work with other systems that list the directory on the screen. The AERCO system has an auto boot program which loads this program as soon as the computer and disk drives are turned on.

Line 10 just sets up the colors
Line 20 sets the variables for the starting position of the cursor.
Line 30 loads the directory and jumps to line 50
Line 40 loads your program from the directory
Line 50 to Line 170 prints and moves the cursor to the start of each listing
Line 130 checks for the ENTER key and jumps to 180 when pressed
Line 180 locates the first space after the first quotation marks in line 40
Line 190 to Line 210 pokes the program listing pointed to by the cursor into line 40, using CODE SCREEN\$
Line 220 jumps to 40 and loads your program

There is a maximum of 14 characters permitted in the listing. 10 for the program name and 4 for the period and type. This program will continue to run if you are only loading BIN type programs, but is lost when a basic program is loaded. The program may be changed in any way after line 40. If any changes in length of program lines are made before 40 you must recalculate the offset of 82 in line 180.

Sincerely,
Lee Freedline
Homer City, PA

```
10 BORDER 0: PAPER 0: INK 7
20 LET P=5: LET Q=0
30 CAT "": GO TO 50
40 CAT " "
50 PRINT AT (P+1),Q;" "
60 PRINT AT (P-1),Q;" "
70 FLASH 1: PRINT AT P,Q;">"
80 FLASH 0: LET A$=INKEY$
90 IF A$="5" AND Q=16 THEN PRINT AT P,Q;" ": LET Q=0: GO TO 50
100 IF A$="6" THEN LET P=P+1
110 IF A$="7" THEN LET P=P-1
120 IF A$="8" THEN PRINT AT P,Q;" ": LET Q=16: GO TO 50
130 IF A$=CHR$ 13 THEN GO TO 180
140 IF P<5 THEN LET P=5
150 IF P>15 THEN LET P=15
160 FOR f=1 TO 5: NEXT f
170 GO TO 50
180 LET Z=(PEEK 23635+256*PEEK 23636)+82
190 FOR A=1 TO 14
200 POKE Z,CODE SCREEN$ (P,Q+A)
210 LET Z=Z+1: NEXT A
220 GO TO 40
```


MORE ON TS DISK DRIVES

By Chuck Dawson

Last time, I reviewed the Timex of Portugal Disk Drive System for the TS-2068. Now that I have had some time to work with the system and try it with several programs, I thought an update of the review might be in order. My overall impression? I love it. Programs load in a flash. Sometimes you don't even get to look at those nice "NOW LOADING" screens long enough to read them. I have a program called "ARTWORX" that takes three minutes and thirty-five seconds to load from tape. From disk, it takes 22 seconds. That is 10 times as fast! The question of which programs can be put on disk is best answered this way: If you can make a tape backup copy of the program, then you can transfer it to disk. You have to give a machine code portion a name because when the basic part of the program tries to load the machine code, a LOAD * "" CODE won't work. However, a LOAD * "name" CODE will. Other than that, all that is required is the addition of the * after LOAD to signify that the computer is to look on disk for the program instead of on tape.

I had quite a bit of trouble with scrambled files and bad SAVES until I figured out that the location of the disk right under the TV was at fault. It seems that some TVs put out a fairly strong signal from their local oscillator or power supply and these signals can confuse the disk controller. Moving the disk system to a location beside the TV rather than right under it fixed the problem and I have had no more trouble. I understand Zebra Systems is including with each purchase instructions as to the best place to locate the drive.

Also, the order in which you turn on the equipment is important. Turn on the disk power supply first, then the computer. When done the other way, a power spike is produced which could knock out the 2068. If you just want to use the computer without the drive, you can leave the drive off and turn on the computer, but I don't think many people will want to operate without the drive once they have it.

One really nice feature of The Operating System (TOS) is the start up file that automatically loads and runs whenever the 2068 is turned on or reset (yes we now have a reset button). I have my printer driver program as a start up file on disks that make use of the printer. For instance, one disk holds a stock portfolio program which allows a printout of data. For the full sized printer to operate, it has to have a driver program installed. So, whenever the Stocks disk is inserted and the system reset, the driver automatically loads with no effort on my part. This keeps forgetful people like me from adding a lot of new data to the program only to find out it cannot be printed because I forgot to load the driver.

By the way, this is the type of program where the disk system really shines. Any program which requires data to be added a little at a time over a long period is a pain with tape because the new version has to be resaved each update session. With the disk system, the program itself can give the SAVE command at the proper time and the save takes place so quickly, one hardly notices except for the low hum and click of the drive. Like I said, I love it.

TIMEX TIPS
by Chuck Dawson

QUESTION: I understand the C-64 computer has a built-in clock that can be set by POKEing in half a dozen values prior to running a program, and the time can be accessed to interrupt a program. Does the TS-2068 have this feature?

ANSWER: Yes, and you only have to POKE three values. The "clock" is updated 60 times each second at location 23672. When this number reaches 255 (about four and a quarter seconds), it is reset to zero and the number in location 23673 is incremented. When it reaches 255 (about 18 minutes), it is set to zero and the number in location 23674 is incremented. That gives us over 77 hours before the clock repeats. The easiest way to use the "clock" is to POKE the three locations with 0 at the beginning of your program. That way, you start at a known value of zero.

Now, let's say you want the program to loop doing some task until 25 minutes have passed, and then branch to an alarm routine. Somewhere in the loop, place the following line:
100 IF PEEK 23672+256*PEEK 23673+65536*PEEK 23674>90000 THEN
GOTO 3000

The number 90000 is calculated by $60 \times 60 \times 25$. Use the ">" sign instead of equals because the chances of the program calculating the value at the moment it is exactly equal to 90000 is pretty small.

Another variation on this theme might be a game that several people play, each trying to beat the other's best time. Again set the three clock locations equal to zero at the beginning of the game and then at the end, calculate the seconds used by the following line:

```
1000 LET SECONDS = INT((PEEK 23672+256*PEEK 23673+65536*PEEK  
23674)/60)
```

By the way, the "clock" is stopped during SAVE and LOAD.

QUESTION: I know that if a number is calculated and the result is a whole part and a line of digits after the decimal, I can strip away the fractional part with INT. But I want whole numbers and two digits after the decimal (like dollars and cents). How can I do this?

ANSWER: Easy! Just multiply the number by 100, take the INTegar part, and divide the answer by 100. Let's say the answer to your calculation is contained in the variable A. Just add the following line to your program prior to printing:

```
100 LET A = .01* INT(100*A)
```

You may want to round the answer to the nearest cent. In that case the line would look like this:

```
100 LET A = .01* INT(.5+100*A)
```

I hope this helps. Good luck.

We started this column in the Summer, 1985, issue with an article on the use of the function SCREEN\$ to identify a character in the PRINT POSITION, and learned that SCREEN\$ was more or less limited in such use, to those characters having codes between 32, a space, and 127, the copyright symbol. SCREEN\$ can be applied to other characters, and even to graphic symbols, but its response is not complete, and requires that we have an understanding of its behavior under such circumstances, in order properly to use it. Hence there was a short program in the initial article that allowed us to INPUT characters of any kind and to get a printout showing how SCREEN\$ considered that character. The program is repeated below for your convenience along with a printout of the title portion..

```

10 PRINT "CHAR- CODE  SCREEN$
CODE  LEN  "
20 PRINT "ACTER OF  (L,C)
SCREEN$ SCR."
30 PRINT "      CHAR.
(L,C) (L,C)"
40 PRINT "      "
100 FOR n=5 TO 21
110 INPUT a$
120 PRINT AT n,1;a$;TAB 7;CODE
a$;TAB 15;SCREEN$ (n,1);TAB 22;C
ODE SCREEN$ (n,1);TAB 30;LEN SCR
EEN$ (n,1)
130 NEXT n

```

CHAR- ACTER	CODE OF	SCREEN\$ (L,C)	CODE SCREEN\$ SCR. (L,C)	LEN (L,C)

You are encouraged to try various characters in this routine, not only those in the selected CODE range, but inverses, the chunky graphics on the keyboard and UDG characters, both those that are resident in the UDG area on power-up, and some that are of your own POKING. It is this last category that is of particular interest to us today.

If you tried a UDG of your own POKING, you would have learned that CODE SCREEN\$(L,C) and LEN SCREEN\$(L,C) both return a value of zero. This means that SCREEN\$ does not "see" a graphic character at all. It "sees" nothing, not even an empty space; because a space has a CODE of 32 and a LENGTH of one.

On the surface this information on the identification of a character does not appear useful. But it is; because in all other cases CODE SCREEN\$(L,C) returns a value greater than zero. Let us generate a graphic character and put this property to a test.

You may, for this test, use a UDG character that you POKED. If you POKED it into the first available position in the UDG section, it can be referred to, in a program, as CHR\$ 144. Or you might like to create a graphic character in a quicker fashion, and identify it as a\$. For example.....

```

5 LET a$=CHR$ 35+CHR$ 21+CHR$
1+CHR$ 8+CHR$ 140

```

There are other and shorter ways to write this line, but the CODE numbers help to explain what is going on when you RUN it. CHR\$ 35 is the "@" symbol. CHR\$ 21 + CHR\$ 1 are equivalent to OVER 1, and CHR\$ 8 says to back space the character that follows. CHR\$ 140 is one of the chunky graphics. Now ENTER...PRINT a\$. How do you like the result?

You can get all sorts of unusual symbols with this line simply by changing the first and last terms of a\$. Bear in mind that a\$ is a string, and as it appears above, it has a CODE of 35, for the first character in the string, and a LENGTH of 5, one for each CHR\$ term. You can verify this by ENTERING PRINT CODE a\$ and PRINT LEN a\$. But SCREEN\$ does not see this symbol in the same way and returns a zero for both CODE and LEN. In our work we are interested only in how SCREEN\$ sees the CODE of a character.

Now we are ready to demonstrate this use of SCREEN\$ in our usual routine of FOUR-DIRECTION MOVEMENT. Add the following lines to the one that initiated a\$.....

```

8 FOR n=1 TO 35
12 PRINT AT 21*AND,31*AND;a$
16 NEXT n
18
20 LET L=10: LET C=15
30 LET LL=L: LET CC=C
40 LET L=L+INKEY$="6" AND L<2
1) - INKEY$="7" AND L<0)
50 LET C=C+INKEY$="8" AND C<3
1) - INKEY$="5" AND C<0)
60 PRINT AT L,C;
70 IF CODE SCREEN$ (L,C)=0 THEN
N BEEP .05,25
80 PRINT "B"
90 PRINT AT LL,CC;" " AND (LL<
) L OR CC<)C)
100 GO TO 30

```

RUN the program and move the puck, an inverse asterisk, about with the arrow keys. Whenever the puck hits a graphic symbol, you should get a confirming BEEP. This may not be exciting, but have patience. We are building up to a more involved application. Line 70 asks SCREEN\$ to look at the block of the screen defined by L and C and to BEEP if the CODE of the character there is zero. The only character whose CODE is zero to SCREEN\$ is the graphic symbol.

But, suppose we have a program in mind that has two or more graphic symbols for SCREEN\$ to examine. We can differentiate between symbols by introducing another condition, say COLOR. To demonstrate this, return to the preceding routine and change line 12 to read.....

```

12 PRINT INK 3 AND AND).5;AT 2
1*AND,31*AND;a$

```

Don't run away! All this change does is to add magenta color to about half of the target symbols that will print on the screen. The others will be in black INK as before. Now the ATTRIBUTES of the symbols have different values. The black ones have an ATTRIBUTE value of 56, which is the normal or default value, and the magenta ones have an ATTRIBUTE value of 59. ATTRIBUTES were discussed at length in the Fall, 1985, issue of Quarters.

Now replace line 70 with the following two lines that take ATTRIBUTES into consideration..

```

75 IF CODE SCREEN$ (L,C)=0 AND
ATTR (L,C)=56 THEN BEEP .05,25
76 IF CODE SCREEN$ (L,C)=0 AND
ATTR (L,C)=59 THEN BEEP .05,10

```

RUN the program again. Now when the puck strikes a black symbol it will BEEP in a higher pitch than when it strikes a magenta one. Thus we can differentiate between them.

In future articles there will be a series of short game programs that will further illustrate these concepts. For the most part, they will be bare-bones programs that you can modify and embellish to suit yourself.

REVIEW: Zebra OS-64 Sixty-Four Column Cartridge
by Chuck Dawson

You may have read in the TS-2068 manual about the 64 column capability of the computer but never got it to work. Welcome to the club! Until now, with the possible exception of the MSCRIPT word processing program, no one could get it to work properly. Now, Zebra Systems of Woodhaven, New York, has introduced the OS-64. It is a cartridge that fits inside that little door on the 2068. When you power up, there it is! It is that easy. You can PRINT AT; TAB; LPRINT; and LLIST in 64 columns. LPRINT and LLIST go to a full sized 80 column printer. The interface driver is built in. COPY will work if you have an EPSON (or compatible) printer.

The limitations are not those of the cartridge but of the computer's built in operating system. When 64 columns are in use, you can have only one paper color and the border will match. No local color assignments are allowed. The whole screen will be one paper color and one ink color. The ink is automatically chosen by the system. Blue paper gives you yellow ink. White paper gives black ink. The combinations are fixed. This has consequences you might not think of right off hand. For instance, when LOADING, you don't get that striped border to show how things are going. The picture just sits there with no indication that anything is happening. I hope you have a VU needle or a light on your recorder to tell you when to shut the thing off. Also, the memory map is changed. That means that many programs will not work without modification. Most machine code is placed high in memory to keep it out of the way, but with 64-column in place, this area is used for stacks and system variables that will cause a crash when overwritten. I tried it with Smart Term II hoping to get a 64 column display when receiving text via the modem, but it didn't work. Some Smart Alec will probably rewrite it and make it work but it won't be me.

Strangely enough, PLOT, DRAW, and CIRCLE work as they do in 32 column with x in the range of 0-255 and y in the range of 0-175. It has been well written and so far the only bug I have found is the inability to suppress a line feed which is sent along with the carriage return during COPY. I had to change the switch in my printer to get a proper COPY. They did provide a way to center the COPY on the paper so I guess we came out even on that one. They mention in the instructions that you will probably want to use a monitor instead of a TV. Well, I use a TV and I think the quality of the display is very acceptable. So, in summary, for many applications (where color is not important but lots of text is) this cartridge is just what we have been waiting for.

T.O.S. DISK DRIVE "START" PROGRAM
by Chuck Dawson

If you own a Timex of Portugal Disk Drive system (also called the Zebra FDD system), you may have noticed in the instruction book that any program named "START" will automatically LOAD and RUN when the system is first powered up or the RESET button is pushed. That seems like a good feature but brings up the question: What program should I name START and what should it do?

Here is one possible solution: Use the program to CATALOG what is on the disk and provide a quick and easy way to choose and LOAD a program or file. The program that I came up with is shown on the next page. The screen display is shown at the bottom. The CATALOG command is used to create the name, type, and size of each file on the disk. The "cursor" is the inverse chevron pointing to "BYTE". It responds to shift 6 (the down arrow) and shift 7 (the up arrow). Pressing ENTER will LOAD the file next to the cursor. You don't have to type out the name or remember to add the "*" or anything. In order for this scheme to work, the program must be able to tell the difference between a BASIC program, a machine code file, a SCREEN\$, and so forth. This is done by the suffix following the name. My system is to have all BASIC programs with no suffix or to use ".BAS" after the name. A machine code name is always followed by ".MC". A screen string is always followed by ".SCR". You can make up your own system as long as you are consistent.

The way this all comes together is that the name of the file next to the cursor is stored in A\$. The suffix, if any, is stored in B\$. Lines 180 to 210 place CODE, SCREEN\$, or whatever is needed behind the LOAD command and the file is LOADED. If you stop the cursor next to the name of a DIRECTORY (the bottom name in the example), the program does a GOTO * "DIRECTORY" and starts over, listing the programs in that directory. Modify the program to fit your own files and suffix system. It needs to be SAVED on each side of each disk. Then when you power up or press the RESET button, you automatically get a list of what is on that disk and by moving the cursor, you can quickly LOAD the chosen file.

```

10 REM SAVE "START" LINE 10
20 OVER 0: CLS
30 LET CURSOR=7
40 PRINT AT 1,0;
50 CAT *
60 PRINT AT CURSOR,8;"<"
70 IF INKEY#=CHR$ 10 AND SCREEN$ (CURSOR+1,0)<>" " THEN PRINT AT CURSOR,8;" "
: LET CURSOR=CURSOR+1-(CURSOR=20)
80 IF INKEY#=CHR$ 11 AND SCREEN$ (CURSOR-1,8)=" " THEN PRINT AT CURSOR,8;" "
LET CURSOR=CURSOR-1+(CURSOR=1)
90 IF INKEY#=CHR$ 13 THEN GO TO 110
100 GO TO 60
110 LET A$="": FOR I=0 TO 7
120 IF SCREEN$ (CURSOR,I)<>" " THEN LET A$=A$+SCREEN$ (CURSOR,I)
130 NEXT I
140 LET B$="": FOR I=9 TO 11
150 LET B$=B$+SCREEN$ (CURSOR,I)
160 NEXT I
170 IF B$="DIR" THEN GO TO *A$: GO TO 10
180 IF B$="MC" " THEN LET A$=A$+".MC": LOAD *A$CODE: STOP
190 IF B$=" " THEN LOAD *A$
200 IF B$="SCR" THEN LET A$=A$+".SCR": LOAD *A$SCREEN$: STOP
210 LET A$=A$+ "."+B$: LOAD *A$

```

Name	Typ	Size	Alloc	S	P
DISASSY		9388	10K		P
MOVER		1481	2K		
H-SCROLL		801	1K		P
ZEBRAREM		397	1K		P
RENUMBER		553	1K		P
BYTE	K	1507	2K		P
TEST		354	1K		P
ZEUS	MC	7941	20K		P
BACKUP		1496	20K		P
LOSYS		1501	20K		P
DUMP	MC	473	1K		P
SENDVARS		1105	2K		P
RENUMBER	MC	36	1K		P
DUMP		835	1K		P
Q-SORT		1201	2K		P
RELOC-DR	MC	805	1K		
SORTFILE		904	1K		
TAPE	DIR	5469	8K		

Max 140K Cur 49K Rem 91K

Input of Default Values in Interactive Programs.

Heinz H. G R O T E

In many interactive programs there are values that are generally constant, but shall easily be changeable, if desired. On the other hand it is too troublesome to insert them each run.

Normally the requested value of a variable is typed-in, ending with an ENTER. For default values it is desirable, to hit only the ENTER button to indicate that the programmed default value shall be used. If a different value is desired, this leaves the option of inserting it before typing ENTER.

Since in the BASIC language characters and numbers are stored differently, a method of conversion has to be used when numbers are entered, since ENTER is a character.

If letters are entered, no conversion is needed, since letters are characters.

But there is no built-in conversion from characters to numbers. The number has to be entered as a character string; and then each character has to be converted with the proper weight.

This conversion is done by the subroutine starting at 200. The program from 100 to 150 is used for testing the subroutine.

Positive and negative numbers with a total of 9 characters (digits plus decimal point plus minus sign) can be used. This number can be adjusted by changing the dimension and the initial value of I.

The program was developed on the 2068. Because the characters in the T/S 1000 are arranged differently and no multiple statements are allowed, the program had to be changed slightly. Both versions were tested on the pertinent computer.

If the default values are nonzero, the previous program is not needed if the following method is used with the disadvantage, that it is operationally more complicated.

If a default value is desired, a zero is inputted followed by ENTER. The program branches then accordingly.

CONTINUED . . .

```

100 DIM I$(10)
110 LET DFLT=12.36
120 INPUT "VALUE ? " (DEFAULT
=12.36)
130 GO SUB 200
140 LET XYZ=INPU: PRINT XYZ
150 GO TO 120
200 LET I=10: LET FACT=1: LET I
NPU=0: LET SIGN=1
210 IF I$(1)<>" " THEN GO TO 23
0
220 LET INPU=DFLT: RETURN
230 IF I$(1)="-" THEN LET SIGN=
-
240 IF I$(1)<>" " THEN GO TO 26
0
250 LET I=I-1: GO TO 240
260 IF I$(1)<>" " THEN GO TO 29
0
270 LET INPU=INPU/FACT: IF I=2
AND SIGN=-1 THEN GO TO 330
280 LET I=I-1: LET FACT=1: GO T
O 240
290 LET INPU=INPU+FACT*(CODE I$(
(1)-48)
300 LET I=I-1: LET FACT=10*FACT
310 IF SIGN=1 THEN GO TO 340
320 IF I>1 THEN GO TO 240
330 LET INPU=-INPU: RETURN
340 IF I<1 THEN RETURN
350 GO TO 240

```

OTHER METHOD

```

100 INPUT "VALUE ? " (DEFAULT
=12.36)
110 IF X<>0 THEN GO TO 130
120 LET X=12.36
130 REM CONTINUE

```

```

100 DIM I$(10)
110 LET DFLT=12.36
120 PRINT AT 21.0;"VALUE ?
(DEFAULT=12.36)"
130 INPUT I$
140 GOSUB 200
150 LET XYZ=INPU
160 CLS
170 PRINT AT 0.0;XYZ
180 GO TO 120
200 LET I=10
210 LET FACT=1
220 LET INPU=0
230 LET SIGN=1
240 IF I$(1)<>" " THEN GOTO 230
250 LET INPU=DFLT
260 RETURN
270 IF I$(1)="-" THEN LET SIGN=
-
280 IF I$(1)<>" " THEN GOTO 260
290 LET I=I-1
300 GOTO 240
310 IF I$(1)<>" " THEN GOTO 290
320 LET INPU=INPU/FACT
330 IF I=2 AND SIGN=-1 THEN GOT
O 330
340 LET I=I-1
350 LET FACT=1
360 GOTO 340
370 LET INPU=INPU+FACT*(CODE I$(
(1)-48)
380 LET I=I-1
390 LET FACT=10*FACT
400 IF SIGN=1 THEN GOTO 340
410 IF I>1 THEN GOTO 240
420 LET INPU=-INPU
430 RETURN
440 IF I<1 THEN RETURN
450 GOTO 240

```

Something strange about "VAL"

Heinz H. Grote

If you look at the program below, you are inclined to doubt the good judgement of the author, putting "VAL" all over. But when you program it and compare it with the way, it is normally written, you find that it uses 14 BASIC bytes (12 %) less than the conventional program with 139 bytes.

This is a consequence of the way in which numbers are programmed by the T/S computers. In order to save execution time, the BASIC program contains both, the original number and the normalized floating point equivalent. If you use the VAL-function, this is not the case. You add the 3 codes for VAL and 2 quotation marks but you gain 4 code bytes. (one for the number indicator, and five for the normalized floating point number).

While the running time for the standard program is 5 seconds, the one with the VAL requires 13 seconds.

This example shows, that in order to save 14 memory spaces (12 %) you sacrifice 8 seconds (160 %) execution time.

```

200 LET A=VAL "2.5"
210 FOR I=1 TO 255
220 IF I>VAL "200" THEN GO TO 4
AL "250"
230 LET A=A-I/VAL "2"
240 GO TO VAL "260"
250 LET A=A+I+VAL "1.5"
260 NEXT I
270 PRINT A

```

```

200 LET A=2.5
210 FOR I=1 TO 255
220 IF I>200 LET A=A-I/2
230 GO TO 260
240 LET A=A+1+1.5
250 NEXT I
260 PRINT A

```


Timex / Sinclair Owners...
TAKE A LOOK!



ONLY
\$15 year
SIX ISSUES

**TIME DESIGNS
MAGAZINE**

FOR YOUR
2068 - 1000/1500 - SPECTRUM

- * Readers nationwide (and Canada too!)
- * 30 illustrated Pages
- * Published "On Time"
- * Interesting Articles / Programs
- * Current Information
- * Exclusive "T-S Shopper" Guide

SUBSCRIBE TODAY!

SAVE \$3.00 Off Newsstand Price. Send Coupon (or a copy) with Check or Money Order for \$15 (U.S. Funds Only) to address posted below. We will start your subscription right away upon receipt of your order.

THE DESIGNS MAGAZINE COMPANY
2075 N.W. 10th Ave., Corvallis, Oregon 97331

Name: _____
Address: _____
City/State: _____ Zip: _____

CTM

"...received my moneys worth with just one issue..."

— J. Trenbick

"...always stop to read CTM, even though most other magazines I receive (and write for) only get cursory examination..."

— Fred Blechman, K6UGT

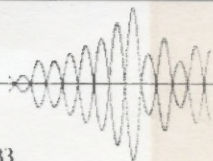
U.S.A.	\$15.00 for 1 year
Mexico, Canada	\$25.00
Foreign	\$35.00 (land) - \$55.00 (air)
(U.S. funds only)	
Permanent (U.S. Subscription)	\$100.00
Sample Copy	\$3.50

CHET LAMBERT, W4WDR

1704 Sam Drive • Birmingham, AL 35235
(205) 854-0271

SyncWare News

P.O. Box 64
Jefferson, NH 03583
Telephone: 603-586-7734



Name: _____
Street: _____
City/State/Zip: _____
Payment:
() check () VISA () Master Card
Card #: _____ Expires: _____

- () Send me 1 year of SYNCWARE NEWS! Enclosed is \$16.95
- () I want the BEST OF SYNCWARE vol.1 All of last year in one big volume. ONLY \$16.95

Do you find yourself looking for good, solid information for your Timex Sinclair computer? Do you want to know what makes it tick? Are you looking for new software, books, hardware or better yet, slicker ways to do your own programming? A YES answer to any of these questions is reason enough to subscribe to the SyncWare News. SWN is not a newsletter, but it is a journal for your machine. We try to explain the logic behind algorithms and not just give a program listing. We have advanced and beginning tutorials in a variety of areas. Our format is changing and upgrading, to provide easier, more appealing reading for everyone. If you're wary of wasting your money, \$2.00 will bring a sample issue for your perusal. We at SWN like our new direction, and we are sure that you'll agree.

Do you have a program or product for sale? Would you like to tell others about a service that you can perform? Drop us a line and we will tell others.

Subscribe !

1 year (6 issues)

\$ 16.95

Add \$3 a year in Canada, all other foreign add \$5 per year.

ACZ SYSTEM LEDGER

"A small business accounting system"

A true double entry system that has the benefits of a manual ledger without the drawbacks. The program identifies transactions that do not balance and gives you the opportunity to make corrections. It will not post out-of-balance transactions, so the books always balance.

1. ACZ GENERAL LEDGER for the TS1000, TS1500 and Sinclair ZX81 all with 16K memory. This version allows 50 Balance Sheet accounts, 50 Income Statement accounts and processes 400 entries and accounts per session. Prints transactions and the three fundamental financial statements: Income Statement, Balance Sheet, and Owners Equity. Originally sold for \$29.95. Now only \$19.95.

2. ACZ GENERAL LEDGER 2.000 for the TS2068. The financial statements feature a flexible design so the reports fit your business whether it is a proprietorship, partnership, or corporation. And there's plenty of capacity with up to 150 named accounts, and up to 800 entries and accounts each month. Prints the following reports: Year-to Date Income Statement, Monthly Income Statement, Balance Sheet, Trial Balance, Ledger Detail, Journal Entries, and the Chart of Accounts. Originally sold for \$39.95. Now only \$19.95.

WMJ DATA SYSTEMS

4 BUTTERFLY DRIVE

HAUPPAUGE, NEW YORK 11788

